Multi-Resolution Planning in Large Uncertain Environments

Leslie Pack Kaelbling
MIT AI Laboratory

with Terran Lane and Georgios Theocharous

Project Agenda

- Near-deterministic abstractions for MDPs
- Near-deterministic abstractions for POMDPs
- Enormous simulated robotic domain
- Demonstrate on real robot
- Teleological decomposition

The Problem

How to select actions in a very large uncertain domain?

- Markov decision processes are a good formalization for uncertain planning
- Optimization algorithms for MDPs are polynomial
- in the size of the state space
- which is exponential in the number of state variables!!

Abstraction and Decomposition

Our only hope is to divide and conquer

- *state abstraction*: treat sets of states as if they were the same
- *state decomposition*: solve restricted problems in sub-parts of the state space
- *action abstraction*: treat sequences of actions as if they were atomic
- *teleological abstraction*: solve restricted problems for sub-parts of the utility function

Hierarchical Uncertain Planning

Given a set of subgoals

- Compute macro actions: optimal strategies for achieving the subgoals
- Compose a policy out of the macros

How to Choose Subgoals?

Given a set of subgoals

- Compute macro actions: optimal strategies for achieving the subgoals
  - time polynomial in size of state space

  ⇒ reduce macros to small subdomains

- Compose a policy out of the macros
  - time polynomial in the number of macros
  - solution quality improves with number of macros (in general)

  ⇒ ??

Near Determinism

Some common action abstractions

- put it in the bag
- go to the conference room
- take out the trash

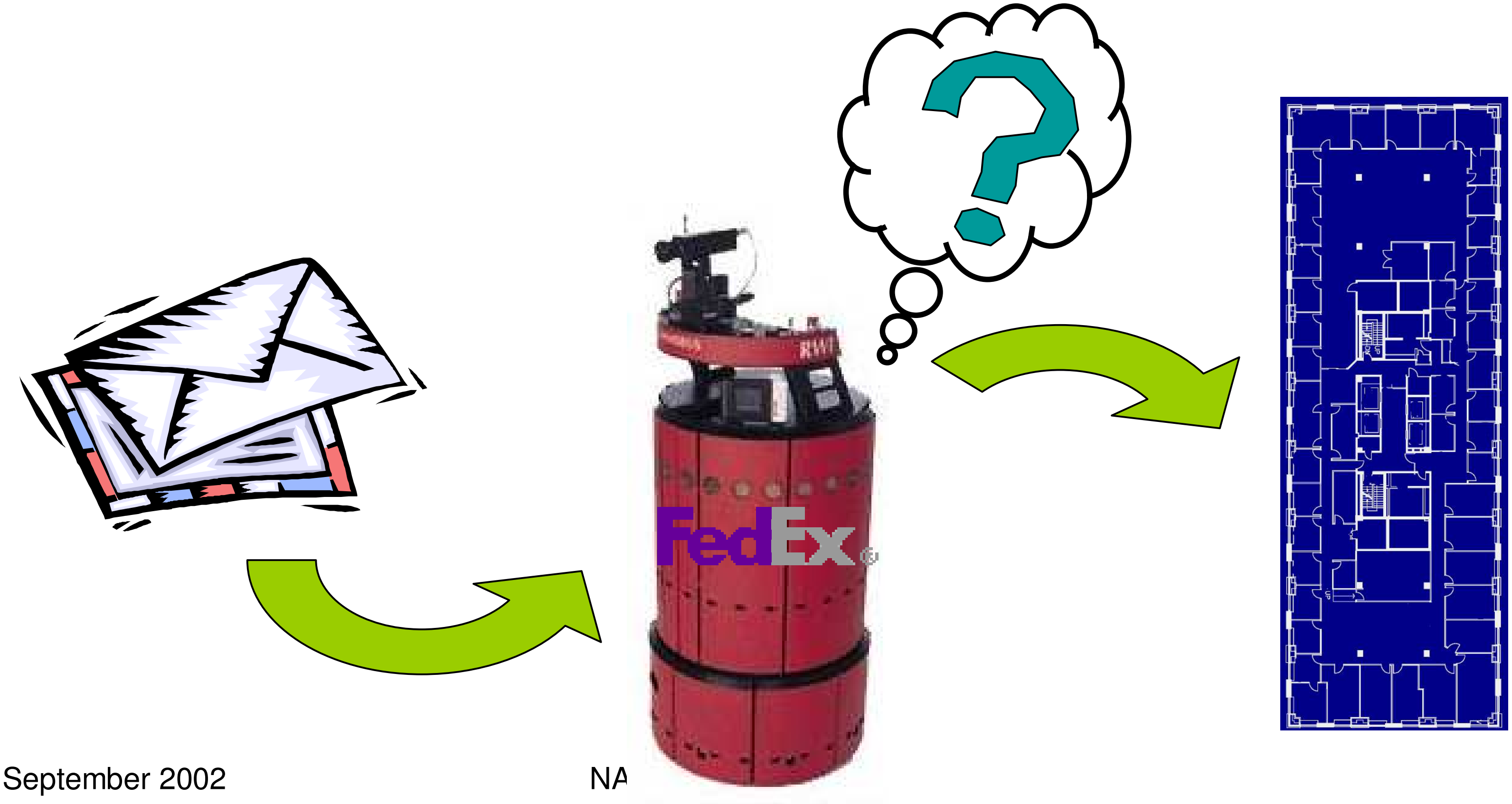
What’s important about them?

- even if the world is highly stochastic,
- you can very nearly guarantee their success

Encapsulate uncertainty at the lower level of abstraction

Sample Domain: Mail Delivery

When it absolutely, positively has to be there…



The target domain

10 Floors

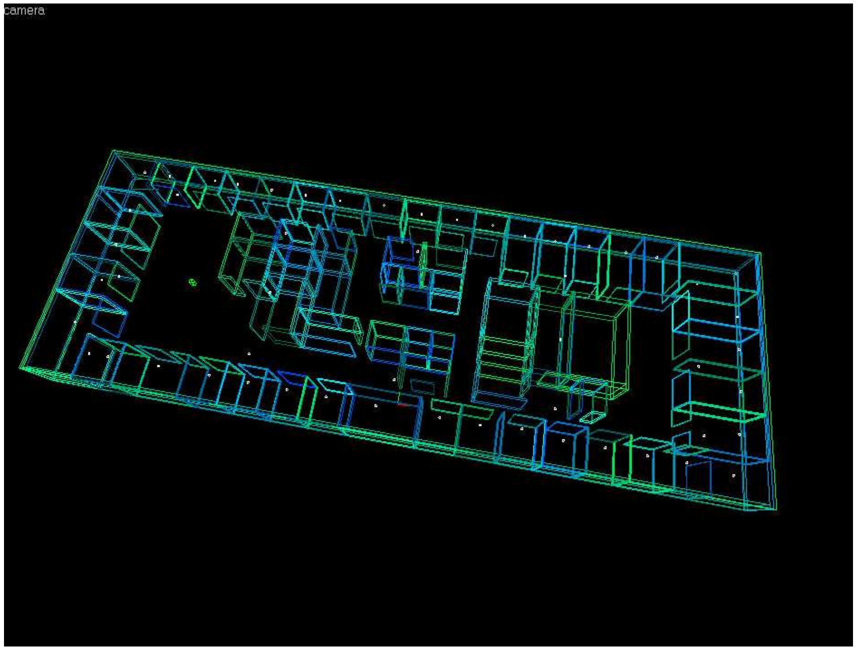
~1800 locations per floor

45 mail drops per floor

Limited battery

11 actions

Total: $|S| > 2^{500}$ states



Two planning problems in one

Problem 1: uncertainty

- Can’t guarantee specific path through world

Solution 1: Markov Decision Process

- Advantage: accounts for uncertainty exactly
- Disadvantage: Doesn’t scale well

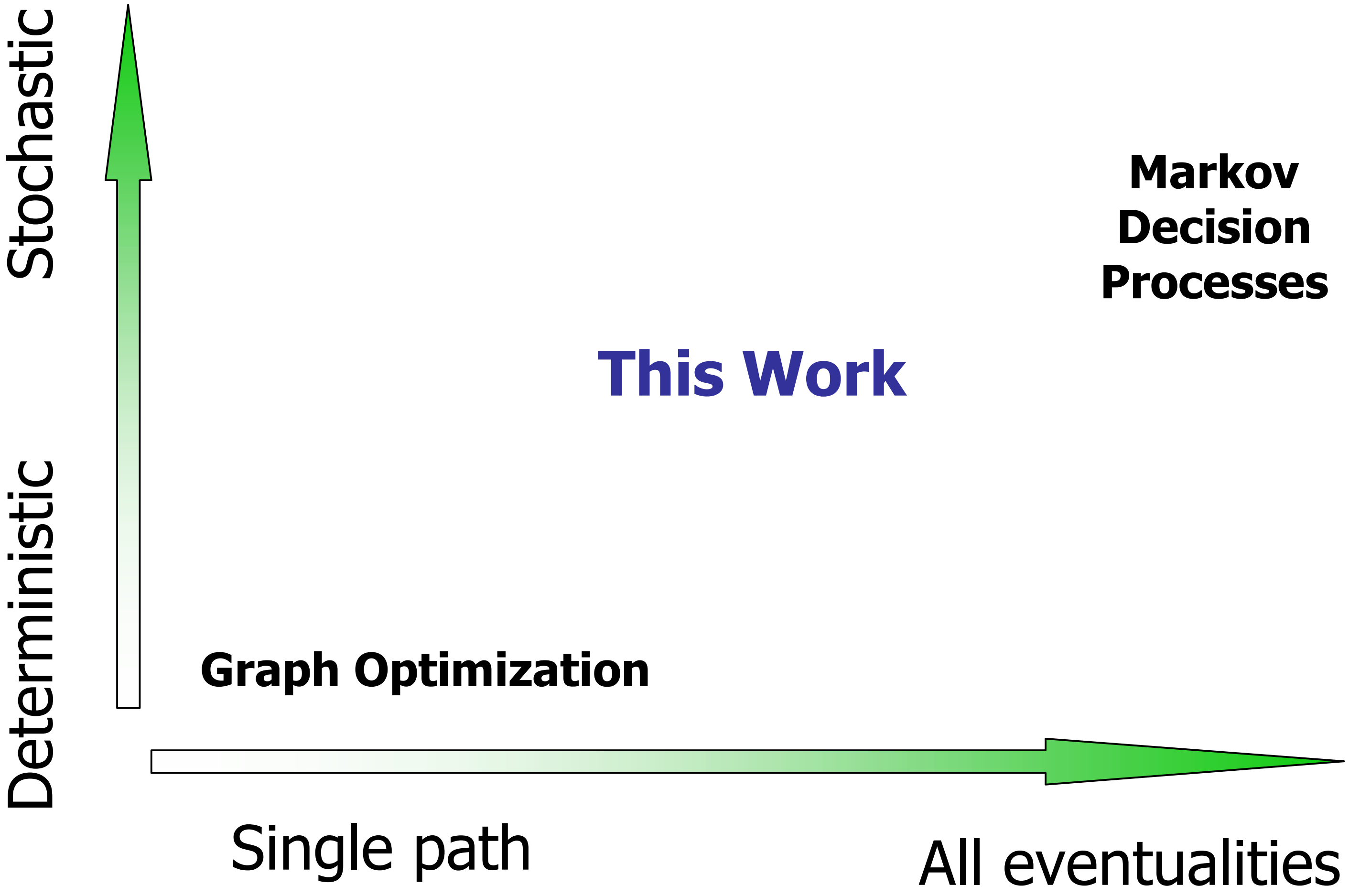
Problem 2: routing

- Path selection combinatorially complex

Solution 2: TSP optimization

- Advantage: scales (relatively) well
- Disadvantage: Doesn’t account for uncertainty

Situating this work



Stochastic

Deterministic

Markov Decision Processes

This Work

Graph Optimization

Single path

All eventualities

A simple example

State space:

- X
- Y
- b (reached goal)

Actions:

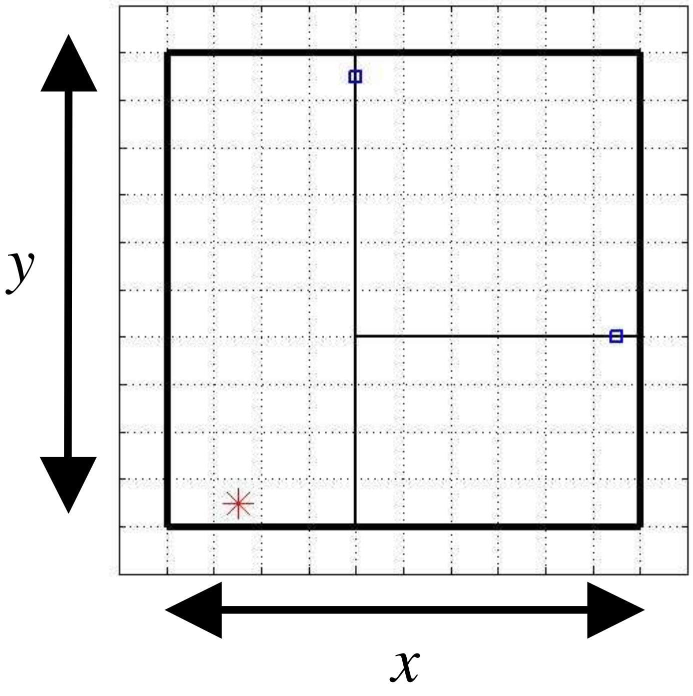
- `N, S, E, W`

Transition function:

- Noise, walls

Rewards:

- $-\varepsilon$/step until $b = 1$
- 0 thereafter



$$|S| = |X||Y|2$$

More destinations

With k destinations we have $|X||Y|2^k$ possible states!

One for each possibly combination of packages that remain to be delivered

Macros deliver single packages

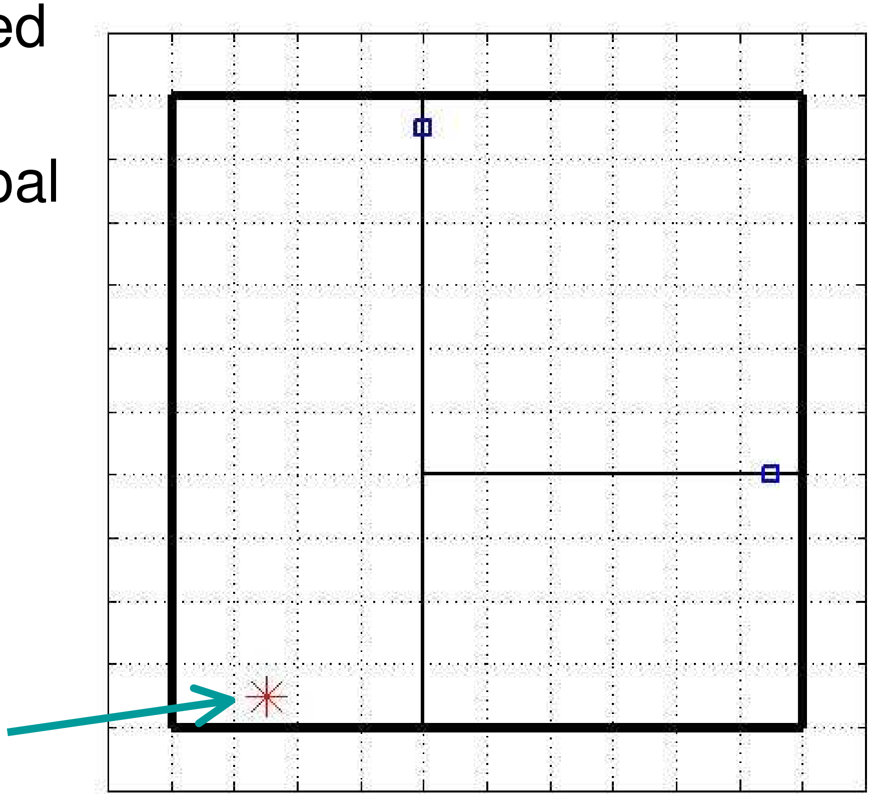
Macro is a plan over a restricted state space

Defines how to achieve *one* goal from any <*x*,*y*> location

Terminates at *any* goal

Can be found quickly

Encapsulates uncertainty



Goal b2

Macros deliver single packages

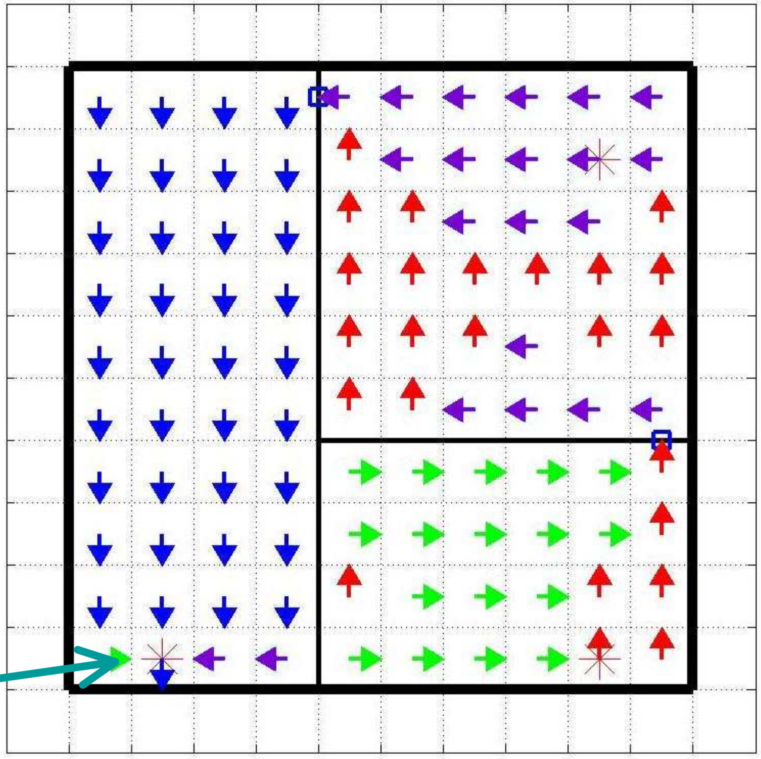
Macro is a plan over a restricted state space

Defines how to achieve *one* goal from any $<x,y>$ location

Terminates at *any* goal

Can be found quickly

Encapsulates uncertainty



Goal b2

Combining Macros

Formally: solve semi-MDP over $\{b\}^k$

- Gets all macro interactions & probs right
- Still exponential, though…

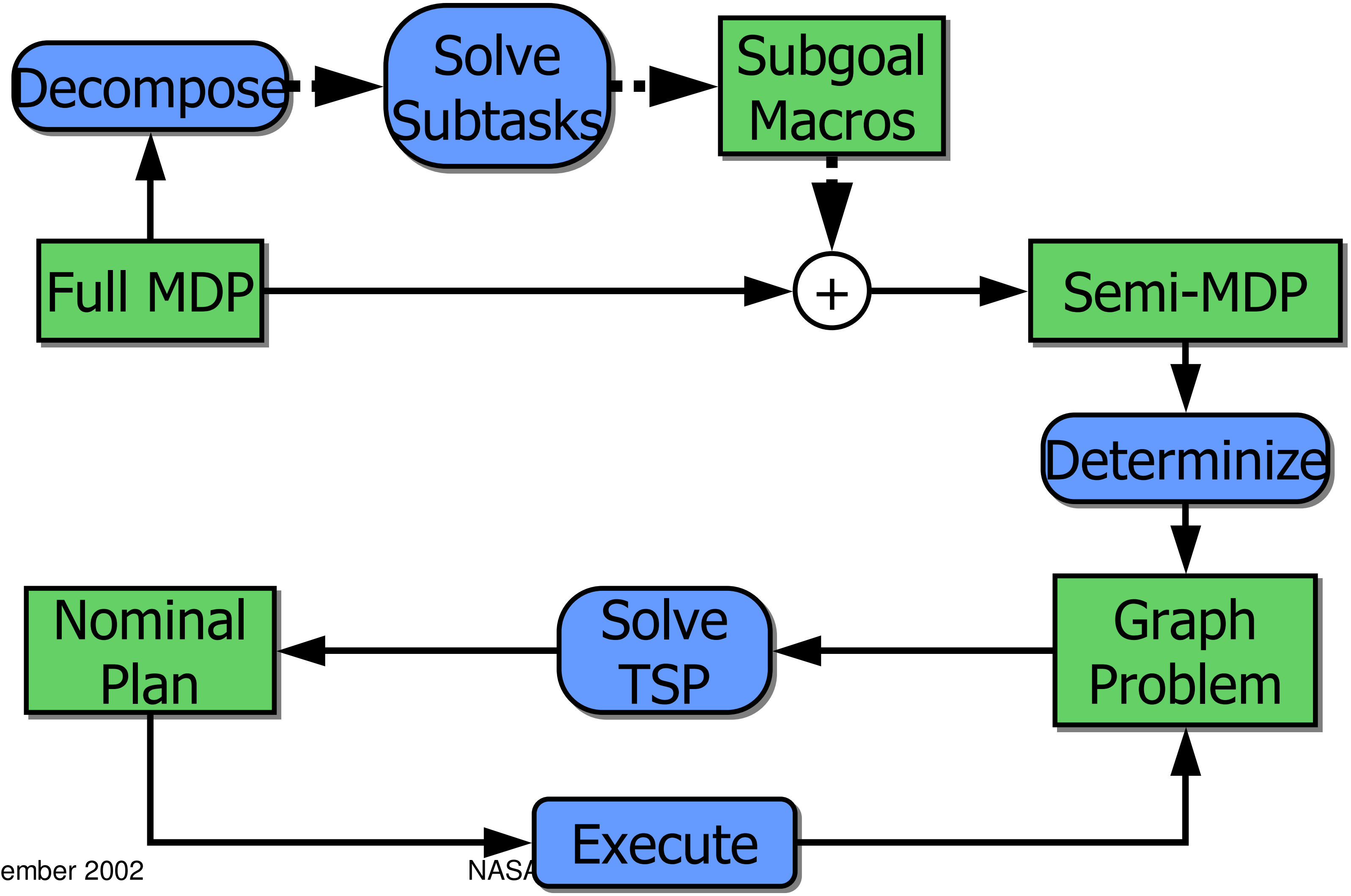
These macros are close to deterministic

- Low prob. of delivering wrong package

Macros form graph over $\{b_1 \ldots b_k\}$

- Reduce SMDP to graph optimization problem

Planner overview



Decompose

Solve Subtasks

Subgoal Macros

Full MDP

+

Semi-MDP

Determinize

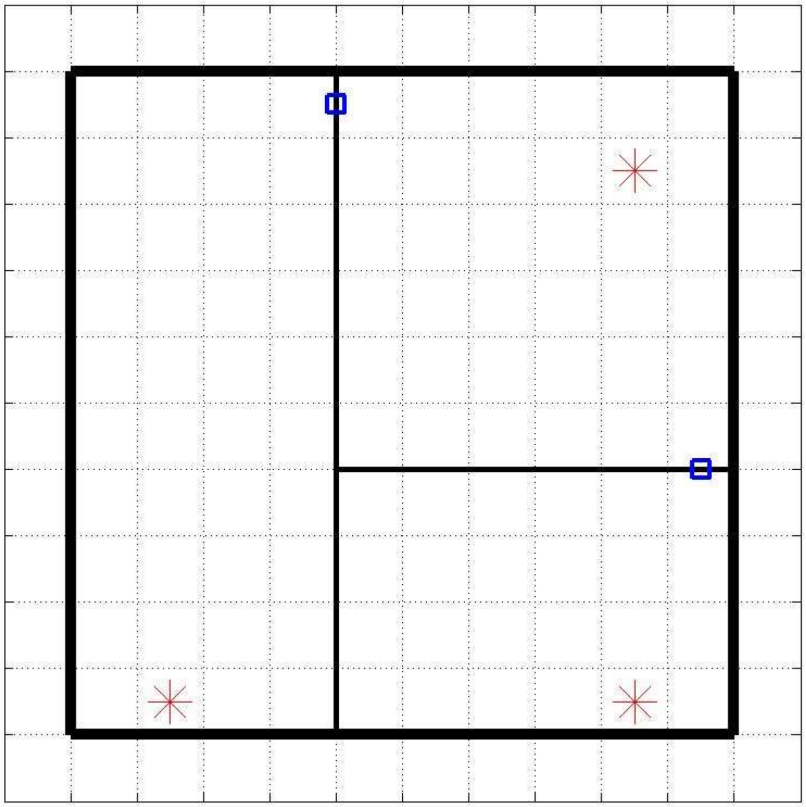
Graph Problem

Solve TSP

Nominal Plan

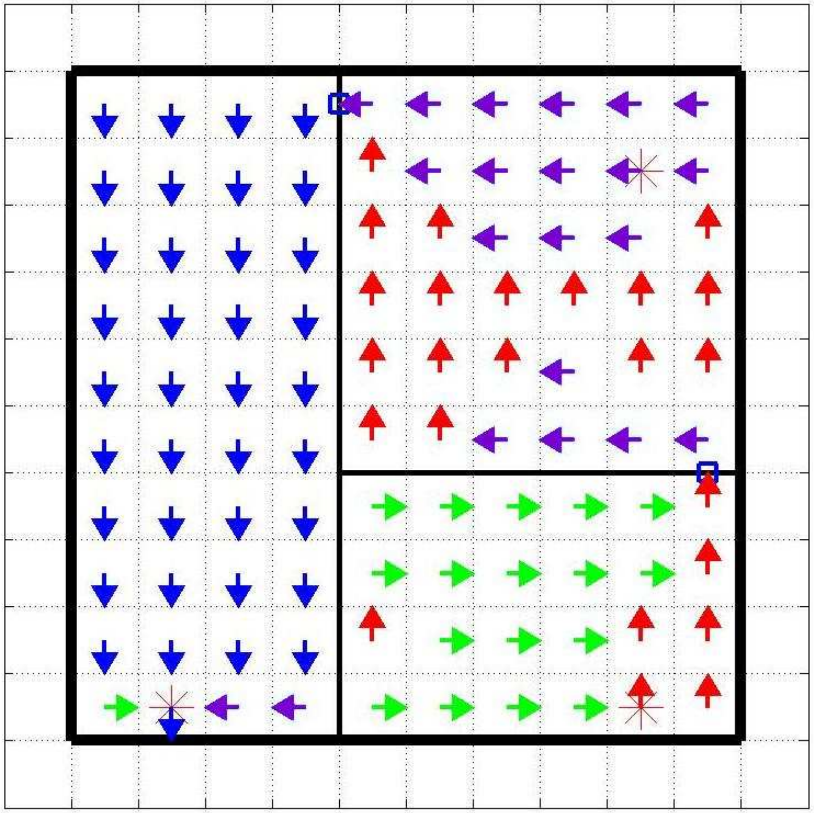
Execute

The algorithm in action



$$|S| = |X||Y|2^k$$

The algorithm in action



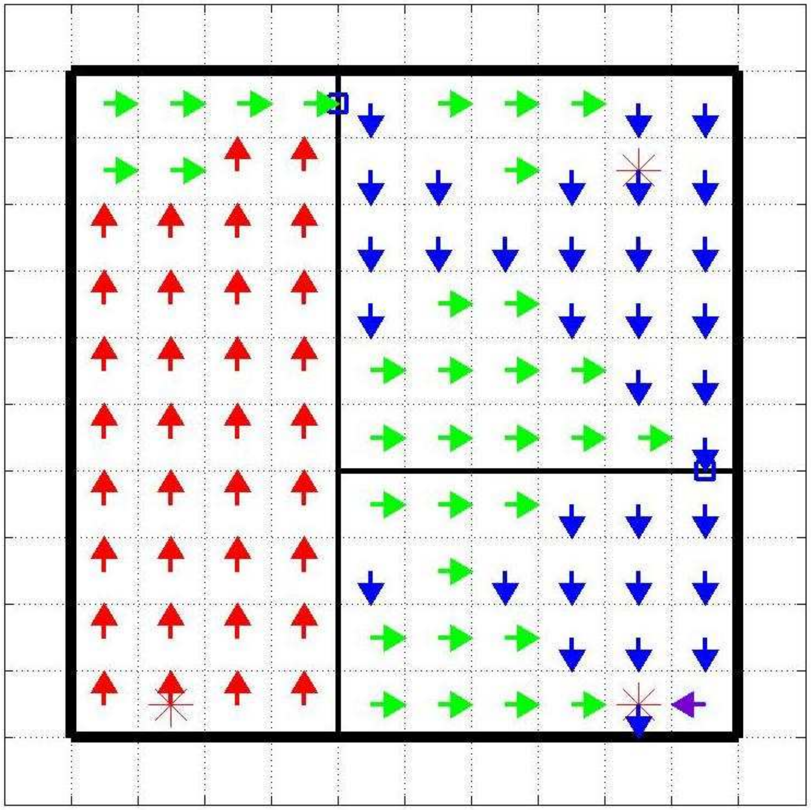
$|S| =$

$|X||Y|$

Time:

$O((|X||Y|)^3)$

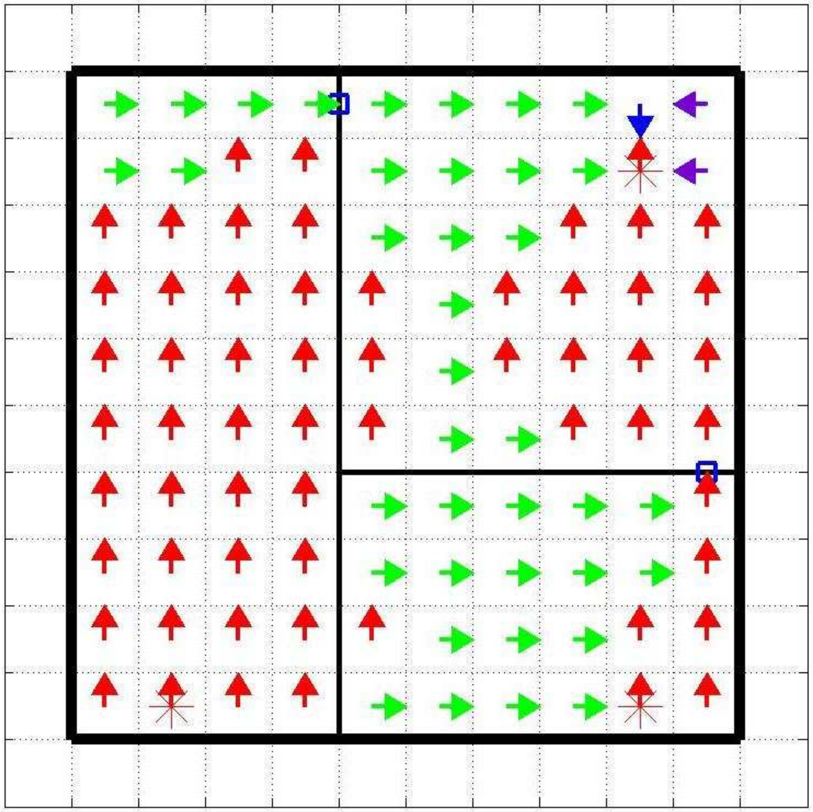
The algorithm in action



$|S| = |X||Y|$

Time: $O((|X||Y|)^3)$

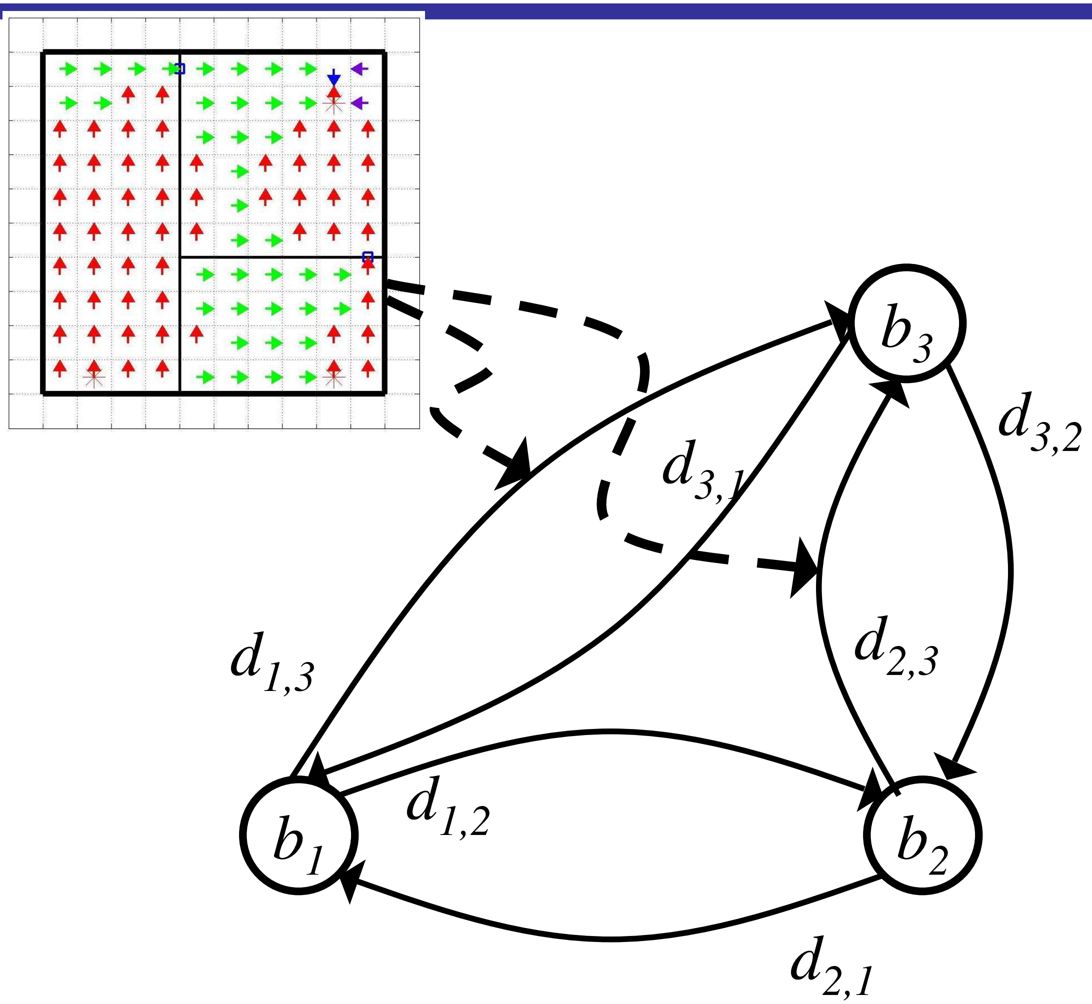
The algorithm in action



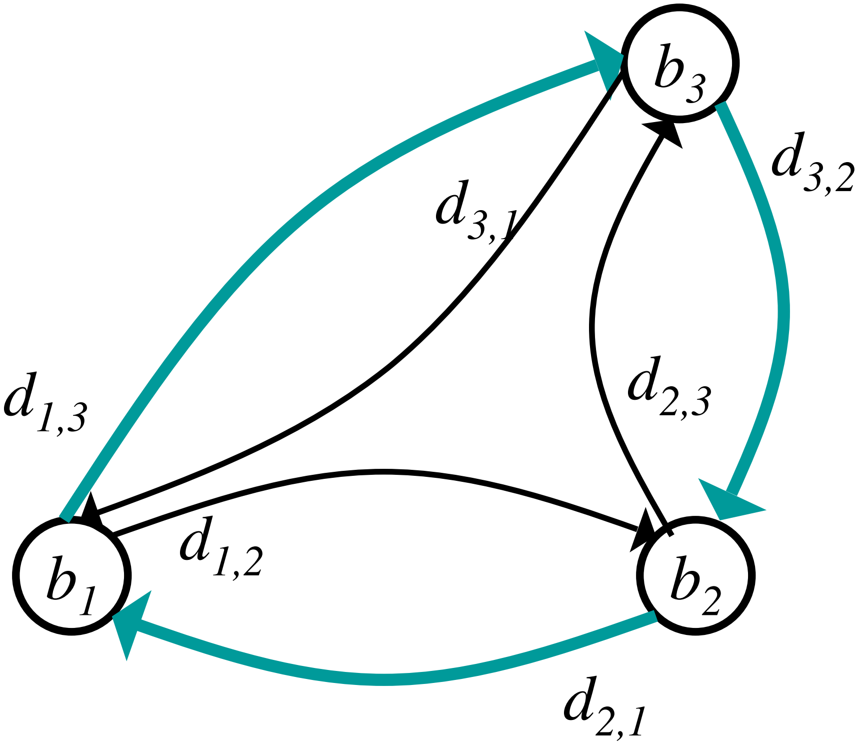
$|S| = |X||Y|$

Time: $O((|X||Y|)^3)$

The algorithm in action



The algorithm in action

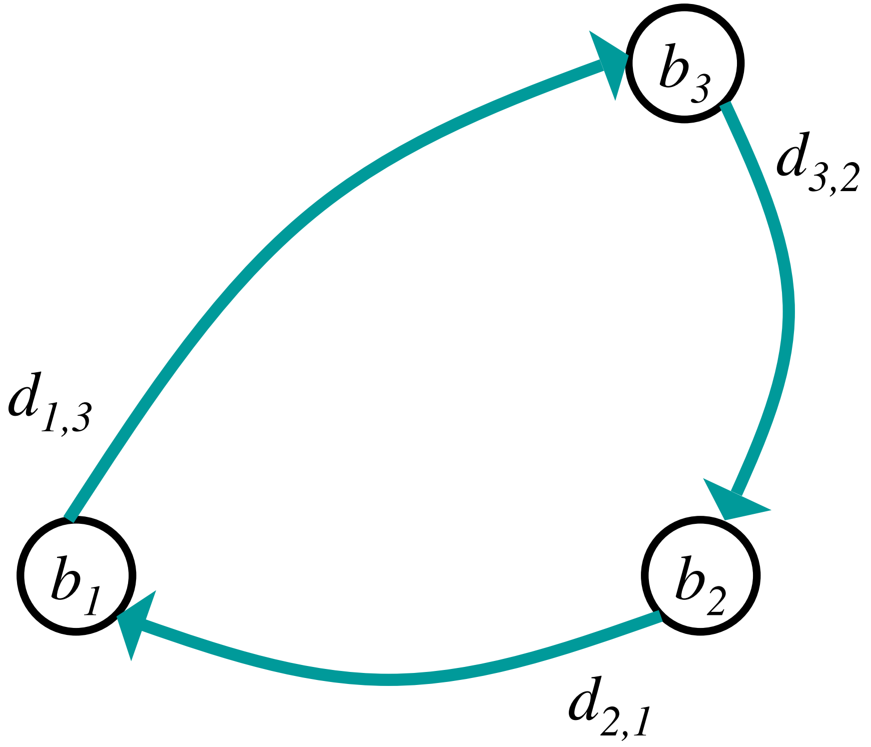


TSP heuristic solver

Time:

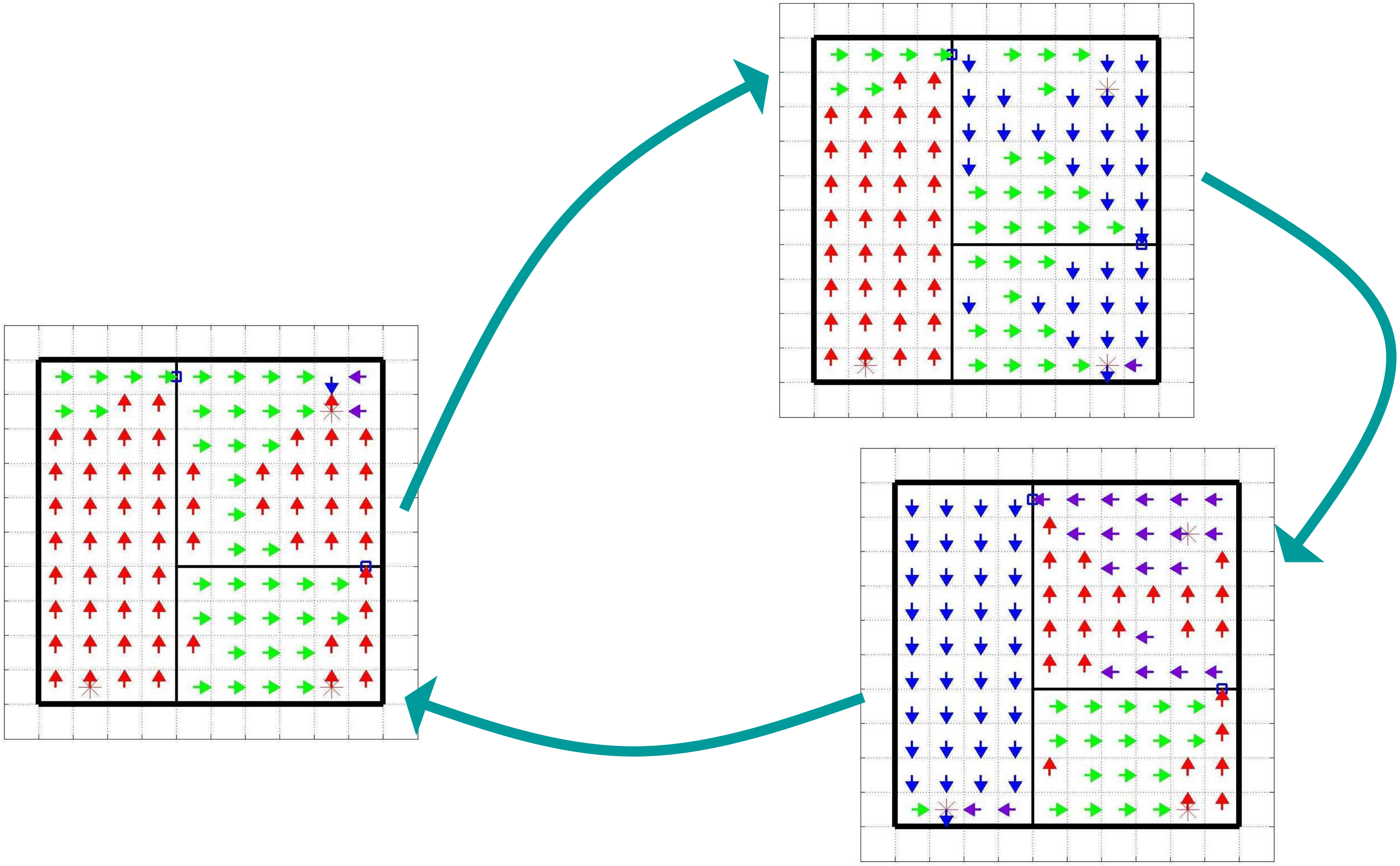
$O(k^c)$

The algorithm in action



Tour defines sequence of MDP macros

The algorithm in action



But does it *work*?

Yes! (Well, in simulation, anyway…)

Small, randomly generated scenarios

- Up to ~60k states (≤6 packages)
- Optimal solution directly
- 5.8% error on avg

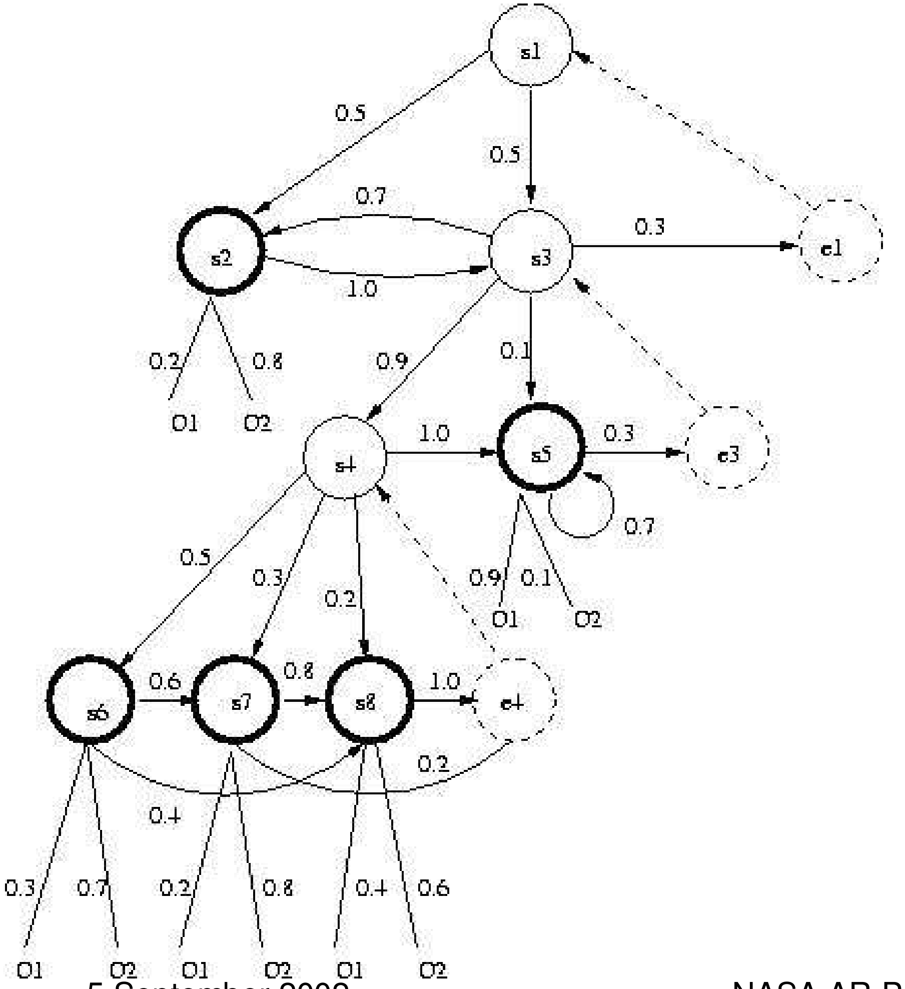
Larger scenarios, based on bldg model

- Up to ~2^{55} states (~45 packages)
- Can’t get optimal soln.
- 600 trajectories; no macro failures
- Theorem gives error bound of 0.3%

Partial Observability

- You can never be sure of the state of the world
- Take uncertainty into account when selecting actions
- POMDP models do this formally
- Wildly intractable, practically
- Hierarchy can help enormously

Hierarchical Hidden Markov Models



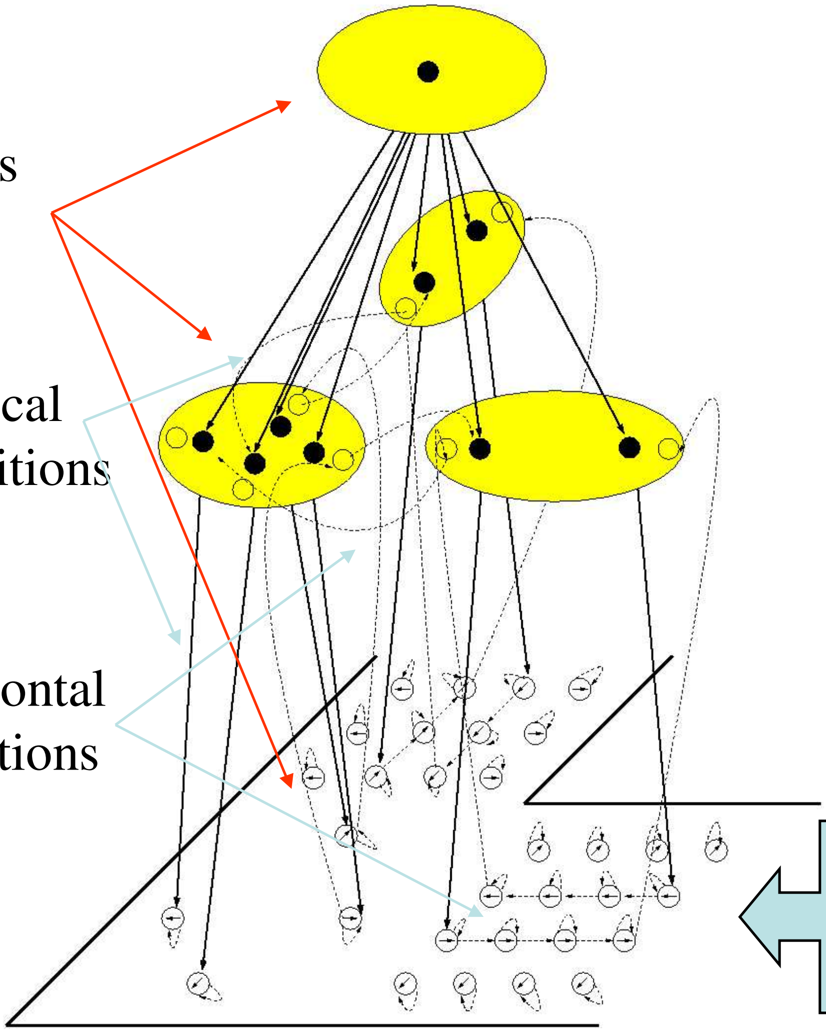
Models hierarchical sequential data

Special case of SCFGs

Past applications:

- Models of natural English text (Fine)
- Identify cursive handwriting strokes (Fine)
- Hierarchical visual tracking of people (Murphy)

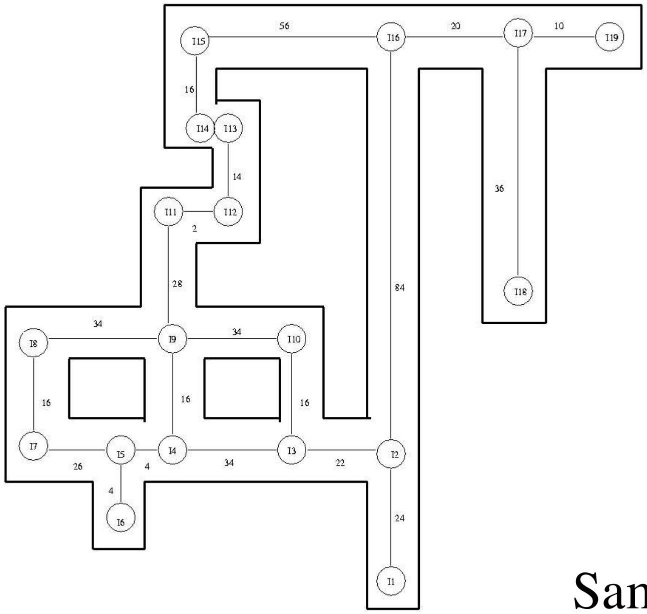
Representing Spatial Environments as HPOMDPs



States

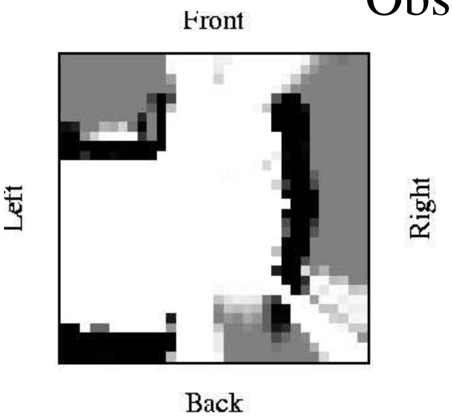
Vertical transitions

Horizontal transitions



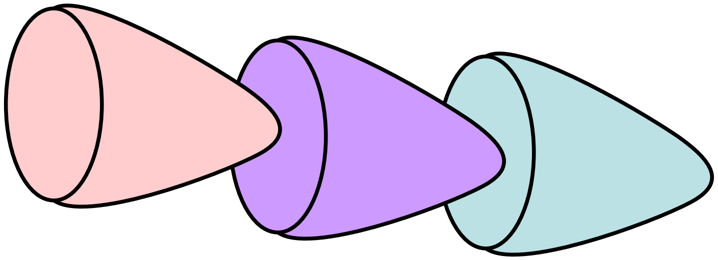
Obser. models

Sample Observation

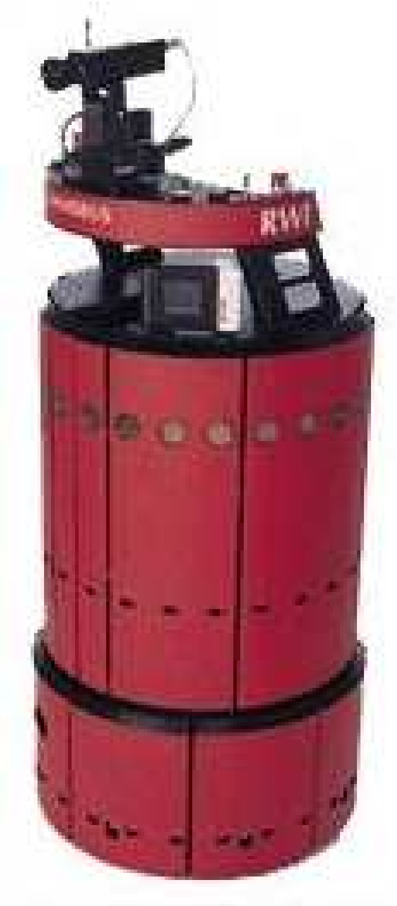


Acting in HPOMDPs

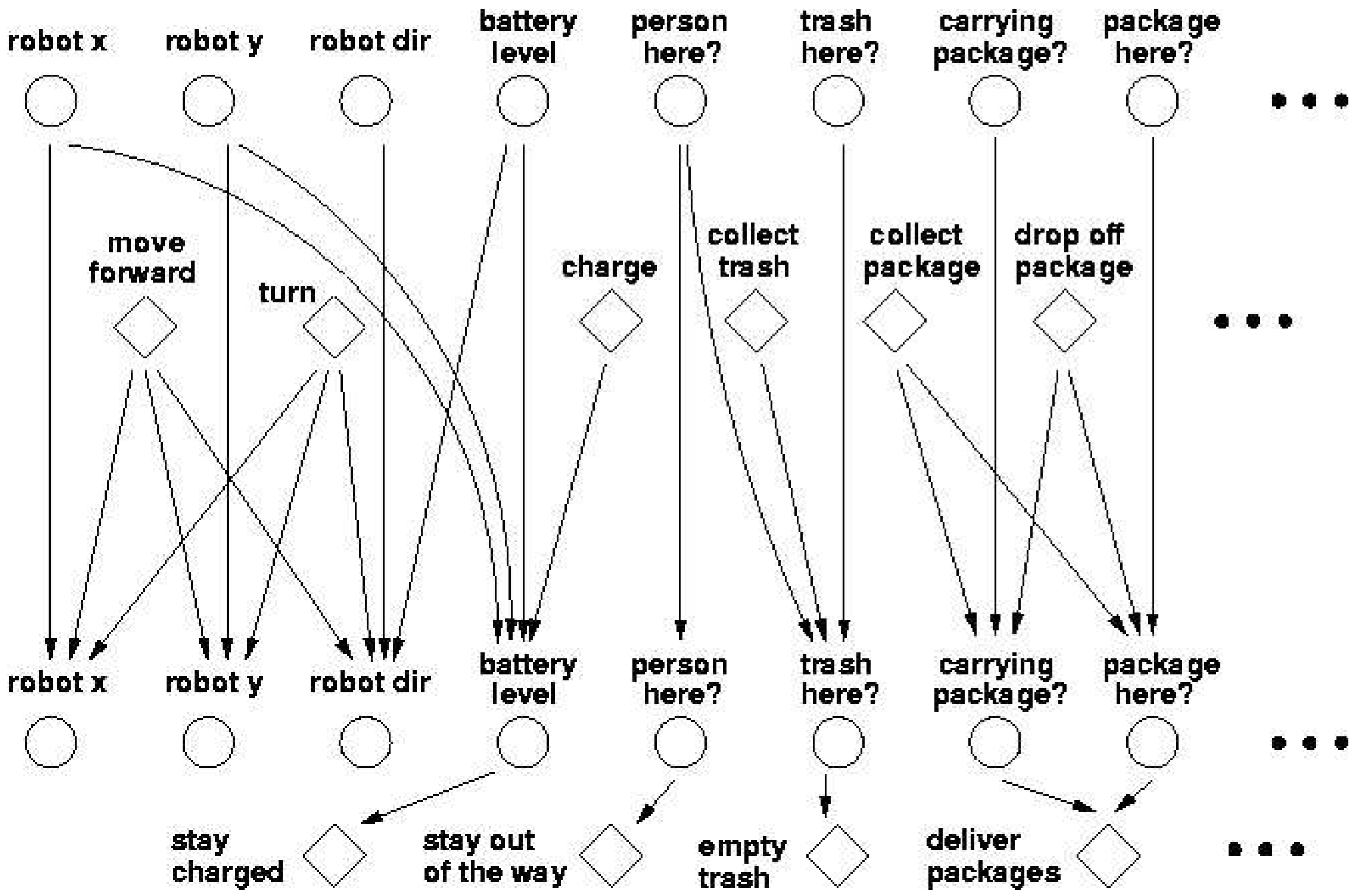
- Previous work on HPOMDPs for state estimation
- Current research project: acting in HPOMDPs
  - macros map belief states to actions
  - choose macros that reliably achieve subsets of belief states
  - “dovetailing”



Port to Real Robot



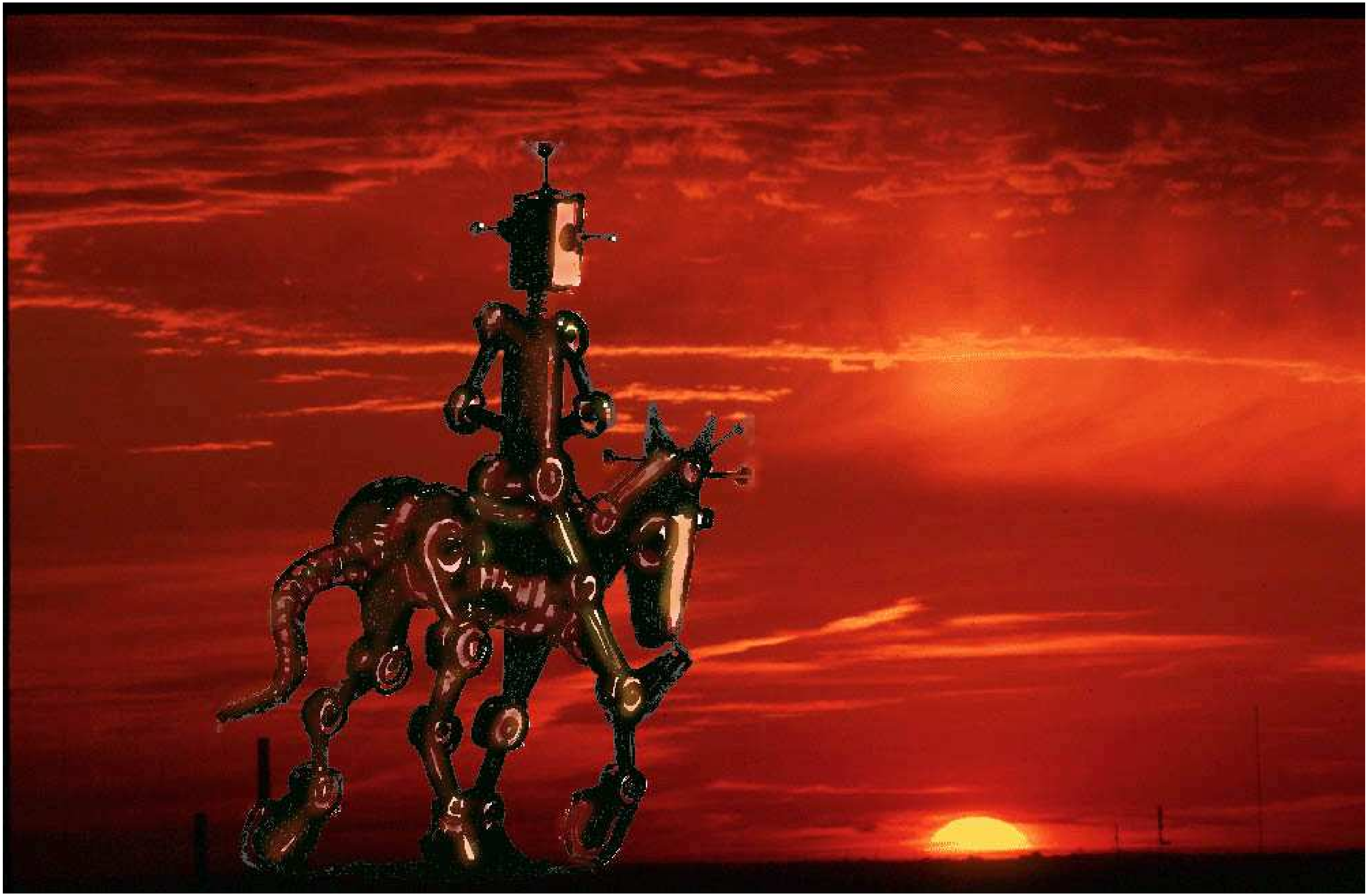
Really Big Domain



Working in Huge Domains

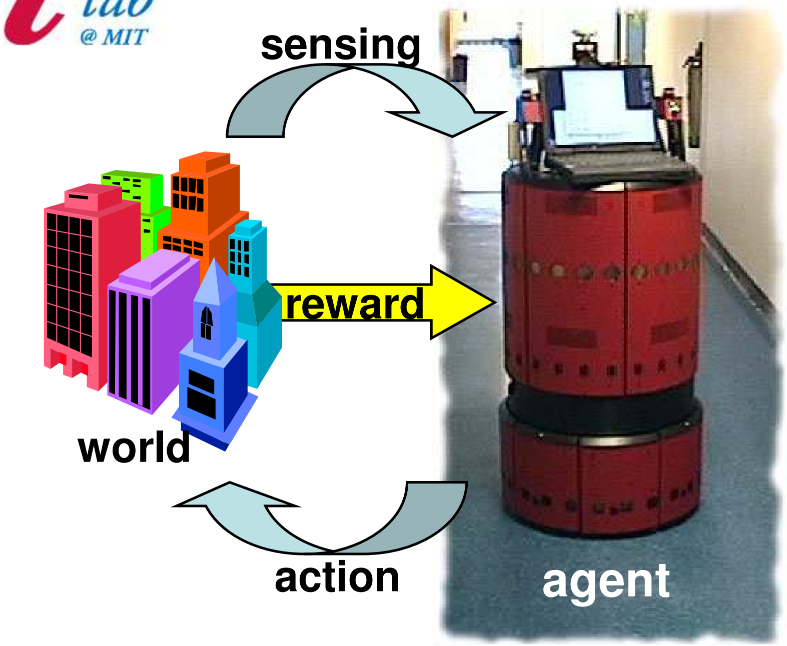
Continually remap the huge problem to smaller subproblems of current import

Decompose along lines of utility function; recombine solutions



Juergen Schmidhuber

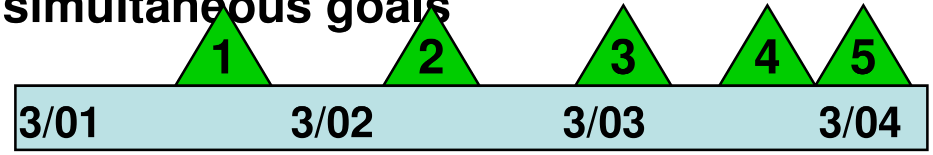
Multi-Resolution Planning in Large Uncertain Environments



Robots capable of extended operations in hugely complex, uncertain multi-objective domains on land and in space

- Solve huge problems through abstraction and hierarchy
- Improve computational performance by seeking near-deterministic abstractions
- Achieve robustness by explicit uncertainty modeling and information gathering

1. Develop near-deterministic abstractions in MDPs
2. Develop near-deterministic abstractions in POMDPs
3. Apply abstraction algorithms in huge simulated robotic domain
4. Demonstrate planning system on real robot domain
5. Develop abstractions based on simultaneous goals


1
2
3
4
5
3/01
3/02
3/03
3/04